

(11)Publication number : 2000-066882
(43)Date of publication of application : 03.03.2000

G06F 9/06
G06K 17/00
G06K 19/07

(71)Applicant : HITACHI LTD
(72)Inventor : YANO HIROHITO
KOBAYASHI MASARU
SASAKI TOSHIRO

[illegible]

[Date of request for examination]
[Date of sending the examiner's decision of rejection]
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]
[Date of final disposal for application]
[Patent number]
[Date of registration]
[Number of appeal against examiner's decision of rejection]
[Date of requesting appeal against examiner's decision of rejection]
[Date of extinction of right]

<http://www19.ipdl.jpo.go.jp/PA1/result/detail/main/wAAAIWay3sDA412066882P1.hi> 2003/12/24

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-66882

(P2000-66882A)

(43) 公開日 平成12年3月3日(2000.3.3)

(51) Int.Cl. ⁷	識別記号	F I	テマコード* (参考)
G 0 6 F 9/06	4 1 0	G 0 6 F 9/06	4 1 0 S 5 B 0 3 5
G 0 6 K 17/00		G 0 6 K 17/00	D 5 B 0 5 8
19/07		19/00	N 5 B 0 7 6

審査請求 未請求 請求項の数6 O L (全 14 頁)

(21) 出願番号 特願平10-232535

(22) 出願日 平成10年8月19日(1998.8.19)

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 発明者 矢野 浩仁

神奈川県川崎市麻生区王禅寺1099番地 株

式会社日立製作所システム開発研究所内

(72) 発明者 小林 賢

神奈川県川崎市麻生区王禅寺1099番地 株

式会社日立製作所システム開発研究所内

(74) 代理人 100095511

弁理士 有近 紳志郎

最終頁に続く

(54) 【発明の名称】 I CカードおよびI Cカードのプログラム管理方法

(57) 【要約】

【課題】 I Cカード内の特定のアプリケーションプログラムやその一部の機能のみを制限可能とする。

【解決手段】 アプリケーションプログラムまたはその機能ごとの実行必須条件をI Cカードに記憶させておき、処理要求があった時に、実行必須条件を読み出し、充足していれば実行可能とし、充足していなければ実行不能とする。

【効果】 従来はI Cカード全体を無効化していたが、本発明では特定のアプリケーションプログラムまたは機能だけを制限でき、他のアプリケーションプログラムまたは機能は無駄にせず利用できる。

(図3)

アプリケーション名	機能名	実行必須条件1			内容
		利用時に 必要	サービス提供時 による実行	内容	
ICクレジットカード	支払い	×	○	○月×日まで有効	—
	入金	×	○	300000円まで入金可能	
△△電子マネー	貯金	○	○	一度に10000円まで	
	引き出し	○	○	無条件	
□□クレジットカード	支払い	×	○	一度に100000円まで	—
××電子マネー	支払い	×	○	○月×日まで有効	

【特許請求の範囲】

【請求項１】 複数のアプリケーションプログラムと各アプリケーションプログラムをそれぞれ実行する前提となる実行条件および１以上のアプリケーションプログラムと各アプリケーションプログラムの各機能をそれぞれ実行する前提となる実行条件の少なくとも一方を記憶したことを特徴とするＩＣカード。

【請求項２】 請求項１のＩＣカードに記憶されているアプリケーションプログラムまたはその機能の実行が要求された時に、当該ＩＣカードに記憶されている当該アプリケーションプログラムまたは当該機能の実行条件を読み出し、その実行条件が充足されているか否かを判定し、充足している場合のみ実行可能とすることを特徴とするＩＣカードのプログラム管理方法。

【請求項３】 請求項１に記載のＩＣカードにおいて、実行可能な複数のアプリケーションプログラムが存在したときの実行優先度を定める情報および実行可能な複数の機能が存在したときの実行優先度を定める情報の少なくとも一方を記憶したことを特徴とするＩＣカード。

【請求項４】 請求項３のＩＣカードに記憶されているアプリケーションプログラムまたはその機能の実行が要求された時に、当該ＩＣカードに記憶されている当該アプリケーションプログラムまたは当該機能の実行条件を読み出し、その実行条件が充足されているか否かを判定し、充足しているアプリケーションプログラムまたは機能が複数あった場合は当該ＩＣカードに記憶されている実行優先度を定める情報を読み出し、その情報に基づいて最優先のアプリケーションプログラムまたは機能を実行可能とすることを特徴とするＩＣカードのプログラム管理方法。

【請求項５】 請求項１または請求項３に記載のＩＣカードにおいて、前記実行条件または前記実行優先度を定める情報の変更許可／変更禁止を定義した変更フラグを記憶したことを特徴とするＩＣカード。

【請求項６】 請求項５のＩＣカードに記憶されている実行条件または実行優先度を定める情報の変更が要求された時に、当該ＩＣカードに記憶されている変更フラグを参照し、変更フラグが変更許可の場合のみ変更可能とすることを特徴とするＩＣカードのプログラム管理方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】 本発明は、ＩＣカードおよびＩＣカードのプログラム管理方法に関し、さらに詳しくは、ＩＣカードに記憶されたアプリケーションプログラムやその機能ごとに実行を制限できるＩＣカードおよびＩＣカードのプログラム管理方法に関する。

【０００２】

【従来の技術】 ＩＣカードは、そのセキュリティの高さのために、定期券、クレジットカード、電子マネー等の

用途に利用されている。これらは、用途にあった機能を盛り込んだアプリケーションプログラムをＩＣカード内に記憶させることで実現している。例えば、電子マネー用のＩＣカードならば、入金、出金、残高照会の３つの機能を持ったアプリケーションプログラムをＩＣカード内に記憶させている。

【０００３】 最近になって、１枚のＩＣカード内に複数のアプリケーションプログラムを記憶させた多目的ＩＣカードが登場してきた（日経マイクロデバイス１９９７年８月号４９ページ）。例えば、１枚のＩＣカード内に定期券のアプリケーションプログラムとクレジットカードのアプリケーションプログラムとを記憶させ、定期券とクレジットカードの両方の用途に利用できるようにしたＩＣカードである。１枚のＩＣカードに複数のアプリケーションプログラムを記憶させることで、例えば異なる交通機関を乗り継ぐ場合でも１枚のＩＣカードで利用可能となるため、利便性が向上する。また、アプリケーションプログラムごとのＩＣカード発行の製造コストを削減できる。

【０００４】 さらに、新たなアプリケーションプログラムや機能を後から追加することが出来るＩＣカードも登場しつつある。

【０００５】

【発明が解決しようとする課題】 ところで、ＩＣカードを失効させる方法がいくつか提案されている（特開平８－５０６３９号公報、特開平８－２６３５７８号公報、特開平７－３１９９６７号公報、特開平５－１３６５０３号公報）。しかしながら、これらの方法はＩＣカード自体を使用不可能とするものであるため、多目的ＩＣカードについては、全てのアプリケーションが使用不可能になってしまう問題点がある。そこで、本発明の第１の目的は、ＩＣカード内の特定のアプリケーションプログラムやその一部の機能のみの実行を制限可能とするＩＣカードのプログラム管理方法を提供することにある。

【０００６】 さらに、交通機関の改札口を通過する際に、ＩＣカードの持つ定期券機能で通過させるのか、料金前払い機能で通過させるのか、金融機関から引き落とす機能で料金精算するのか等、起動可能なアプリケーションプログラムまたはその機能が複数存在する場合に、従来は、実行可能なアプリケーションプログラムまたはその機能をＩＣカードの利用者に知らせ、実行するアプリケーションプログラムまたはその機能を利用者に選択させることにしている。しかしながら、実行するアプリケーションプログラムまたはその機能を利用者に選択させる方法では、手続き処理時間が長くなる問題点がある。そこで、本発明の第２の目的は、実行するアプリケーションプログラムまたはその機能を自動選択可能とするＩＣカードおよびＩＣカードのプログラム管理方法を提供することにある。

【０００７】 さらに、本発明の第３の目的は、ＩＣカー

ド内の特定のアプリケーションプログラムやその一部の機能のみの実行を制限可能とする条件が不当に変更されることを防止できるICカードおよびICカードのプログラム管理方法を提供することにある。

【0008】

【課題を解決するための手段】第1の観点では、本発明は、複数のアプリケーションプログラムと各アプリケーションプログラムをそれぞれ実行する前提となる実行条件および1以上のアプリケーションプログラムと各アプリケーションプログラムの各機能をそれぞれ実行する前提となる実行条件の少なくとも一方を記憶したことを特徴とするICカードを提供する。また、本発明は、上記構成のICカードに記憶されているアプリケーションプログラムまたはその機能の実行が要求された時に、当該ICカードに記憶されている当該アプリケーションプログラムまたは当該機能の実行条件を読み出し、その実行条件が充足されているか否かを判定し、充足している場合のみ実行可能とすることを特徴とするICカードのプログラム管理方法を提供する。上記第1の観点のICカードおよびICカードのプログラム管理方法では、アプリケーションプログラムまたはその機能ごとの実行条件をICカードに記憶させておき、ICカード自身が、又は、リーダーライター側が、実行条件を読み出して実行可能性を判定する。これにより、ICカード内の特定のアプリケーションプログラムやその一部の機能のみの実行を制限できるようになる。

【0009】第2の観点では、本発明は、上記構成のICカードにおいて、実行可能な複数のアプリケーションプログラムが存在したときの実行優先度を決定する情報および実行可能な複数の機能が存在したときの実行優先度を決定する情報の少なくとも一方を記憶したことを特徴とするICカードを提供する。また、本発明は、上記構成のICカードに記憶されているアプリケーションプログラムまたはその機能の実行が要求された時に、当該ICカードに記憶されている当該アプリケーションプログラムまたは当該機能の実行条件を読み出し、その実行条件が充足されているか否かを判定し、充足しているアプリケーションプログラムまたは機能が複数あった場合は当該ICカードに記憶されている実行優先度を決定する情報を読み出し、その情報に基づいて最優先のアプリケーションプログラムまたは機能を実行可能とすることを特徴とするICカードのプログラム管理方法を提供する。上記第2の観点のICカードおよびICカードのプログラム管理方法では、アプリケーションプログラムまたはその機能ごとの実行優先度を決定する情報をICカードに記憶させておき、ICカード自身が、又は、リーダーライター側が、実行優先度を決定する情報を読み出して最優先のアプリケーションプログラムまたは機能を実行可能とすることを特徴とする。これにより、実行するアプリケーションプログラムまたはその機能を自動選択できるよ

うになり、手続き処理時間を短縮できる。

【0010】第3の観点では、本発明は、上記構成のICカードにおいて、前記実行条件または前記実行優先度を決定する情報の変更許可／変更禁止を定義した変更フラグを記憶したことを特徴とするICカードを提供する。また、本発明は、上記構成のICカードに記憶されている実行条件または実行優先度を決定する情報の変更が要求された時に、当該ICカードに記憶されている変更フラグを参照し、変更フラグが変更許可の場合のみ変更可能とすることを特徴とするICカードのプログラム管理方法を提供する。上記第3の観点のICカードおよびICカードのプログラム管理方法では、変更フラグをICカードに記憶させておき、ICカード自身が、又は、リーダーライター側が、変更フラグを読み出して実行条件または実行優先度を決定する情報の変更の許可／禁止を判断し、許可の場合のみ変更を可能にする。これにより、ICカード内の特定のアプリケーションプログラムやその一部の機能のみの実行を制限可能とする条件が不当に変更されることを防止できる。

20 【0011】

【発明の実施の形態】以下、本発明の実施の形態を詳細に説明する。なお、これにより本発明が限定されるものではない。

【0012】一般に、リーダーライターと呼ばれる専用の装置によって、様々な処理要求が、ICカードに対して送信される。ICカードは、前記処理要求に対してセキュリティチェックを行い、それに合格すると、前記処理要求に応じて、ICカード内部の記憶装置に記憶されたアプリケーションプログラムやその機能を実行する。25

30 そして、必要があれば、ICカード内部の記憶装置に情報を書き込み、リーダーライターに情報を送信する。最後に、ICカードは処理終了信号をリーダーライターに送信し、ICカードの処理を終る。また、リーダーライターは、前記処理終了信号を受信し、処理を終了する。

35 【0013】図1は、本発明の一実施形態にかかるICカードの斜視図である。このICカード1は、プレート11に、ICチップ12と、リーダーライターとの情報のやりとりを行う接続部13と、電源を供給する電源供給部14とを設けたものである。前記電源供給部14

40 は、接触ICカードにおける電源端子や、非接触ICカードにおける内蔵バッテリーや発電コイル等を意味する。

【0014】図2は、前記ICチップ12の構成を示すブロック図である。このICチップ12は、記憶装置12mと演算装置12pとを具備し、データバス15により前記接続部13に接続されると共に、電源ライン16により前記電源供給部14に接続されている。前記記憶装置12mには、複数のアプリケーションプログラムと、本発明のプログラム管理方法を実施するための管理プログラムと、前記アプリケーションプログラムまたは

その一部の機能の実行条件を登録した条件テーブル（図3～図5）と、ICカード1の利用者の個人情報等が記憶されている。

【0015】図3～図5は、前記条件テーブルの例示図である。条件テーブルは、アプリケーションプログラム名と、各アプリケーションプログラムの機能と、各機能の実行条件すなわち実行必須条件または実行優先度関数と、実行優先順位とを対応表にしたものである。前記実行必須条件は、当該機能を実行するために充足していなければならない条件を登録した内容と、その内容の変更を利用者に対して許可しているか否かを示す利用者による変更フラグと、前記内容の変更をサービス提供者に対して許可しているか否かを示すサービス提供者による変更フラグとからなっている。前記実行優先度関数は、当該機能の外にも実行可能な機能があった場合に当該機能を優先的に実行するための条件を登録した内容と、その内容の変更を利用者に対して許可しているか否かを示す利用者による変更フラグと、前記内容の変更をサービス提供者に対して許可しているか否かを示すサービス提供者による変更フラグと、当該機能の実行優先度関数がオリジナルな関数であるか否かを示すオリジナルフラグと、複数のアプリケーションプログラムまたはその機能が実行可能であった場合にそれらを実行する優先順位を登録した実行優先順位とからなっている。

【0016】なお、図3～図5の条件テーブルでは、アプリケーションプログラムの機能単位で実行必須条件と実行優先度関数が設定されているが、アプリケーションプログラム単位に実行必須条件と実行優先度関数を設定してもよい。

【0017】図6は、ICカード1の管理プログラムのメイン処理を示すフローチャートである。ステップ101では、リーダーライターからICカード1への呼びかけ信号を受信する。ステップ102では、呼びかけ信号内容が実行条件の設定に関するものか否かを判定し、そうでなければステップ103へ進み、そうであればステップ109へ進む。ステップ103では、呼びかけ信号に対応するアプリケーションプログラムの機能が実行必須条件を満足しているか否かを判定する実行必須条件チェック処理を行う。この実行必須条件チェック処理については、図7を参照して後述する。ステップ104では、すべての機能について前記ステップ103を実行したかをチェックし、実行していないなら前記ステップ101に戻り、実行したならステップ105に進む。ステップ105では、すべての呼びかけ信号について前記ステップ103を実行したかをチェックし、実行していないなら前記ステップ101に戻り、実行したならステップ106に進む。ステップ106では、実行可能な機能がなかったかをチェックし、あったならステップ107へ進み、なかったならステップ108へ進む。ステップ107では、実行可能な機能の中から一つを選択し実行する

機能選択実行処理を行う。そして、メイン処理を終了する。この機能選択実行処理については、図8を参照して後述する。ステップ108では、実行可能な機能が無い旨の送信を行う。そして、メイン処理を終了する。ステップ109では、機能実行条件設定処理を実行する。そして、メイン処理を終了する。この機能実行条件設定処理については、図9を参照して後述する。

【0018】図7は、実行必須条件チェック処理（103）を示すフローチャートである。ステップ201では、ICチップ12の記憶装置12mから、呼びかけ信号が求めているアプリケーションプログラムの機能の実行必須条件の一つを読み取る。ステップ202では、ICチップ12の記憶装置12mやリーダーライターから、前記読み出した実行必須条件の判定に必要な情報を読み出す。ステップ203では、ICチップ12の演算装置12pが、前記読み出した実行必須条件を充足しているか判定する。ステップ204では、前記読み出した実行必須条件を充足していたならステップ205へ進み、充足していなかったらステップ207へ進む。なお、実行必須条件の内容に「無効」が設定されていたら、実行必須条件を充足していないものとする。ステップ205では、呼びかけ信号が求めているアプリケーションプログラムの機能の実行必須条件の全てについて前記ステップ204の判定を行ったかチェックし、行っていないなら前記ステップ201に戻り、行ったならステップ206へ進む。ステップ206では、呼びかけ信号が求めているアプリケーションプログラムの機能を実行可能であると判定する。そして、実行必須条件チェック処理を終了する。ステップ207では、呼びかけ信号が求めているアプリケーションプログラムの機能を実行不能であると判定する。そして、実行必須条件チェック処理を終了する。

【0019】図8は、機能選択実行処理（107）を示すフローチャートである。ステップ301では、ICチップ12の記憶装置12mから、実行可能な機能の一つに着目し、さらにその実行優先度関数の一つを読み出す。ステップ302では、ICチップ12の記憶装置12mやリーダーライターから、前記読み出した実行優先度関数の算出に必要な情報を読み出す。ステップ303では、実行優先度関数の値を算出する。ステップ304では、着目している機能の実行優先度関数の全てについて前記ステップ303の値の算出を行ったかチェックし、行っていないなら前記ステップ301に戻り、行ったならステップ305へ進む。ステップ305では、算出した全ての実行優先度関数の値の和を算出する。ステップ306では、実行可能な機能の全てに対して前記ステップ305の和の算出を行ったかチェックし、行っていないなら実行可能な機能の別の一つに着目して前記ステップ301に戻り、行ったならステップ307へ進む。ステップ307では、実行優先度関数の和が最も高

い機能が複数あるかチェックし、一つしかなければステップ308へ進み、複数あればステップ309へ進む。ステップ308では、実行優先度関数の和が最も高い機能を選択し、ステップ310へ進む。ステップ309では、実行優先度関数の和が最も高い複数の機能の中で実行優先順位が最も高い機能を選択する。ステップ310では、選択した機能を実行する。

【0020】図9は、機能実行条件設定処理(109)を示すフローチャートである。ステップ401では、実行条件の設定を行うための装置から、設定信号を受信する。ステップ402では、設定信号が指定する機能がICカード1内に存在するか否かをチェックし、存在した場合はステップ403へ進み、存在しなかった場合はステップ410へ進む。ステップ403では、設定信号が指定する機能の利用者による変更フラグまたはサービス提供者による変更フラグをチェックし、実行条件の設定を行おうとしている者が変更許可されているならステップ404へ進み、変更禁止されているならステップ410へ進む。ステップ404では、設定信号が指定する機能が実行必須条件か否かをチェックし、実行必須条件ならステップ405へ進み、実行必須条件でない(実行優先度関数である)ならステップ407へ進む。ステップ405では、実行必須条件で新たに設定されようとしている内容が、変更不可能な既設定の実行必須条件と矛盾するか否かをチェックし、矛盾がなければステップ406へ進み、矛盾があればステップ410へ進む。ステップ406では、実行必須条件で新たに設定されようとしている内容が、変更可能な既設定の実行必須条件と矛盾するか否かをチェックし、矛盾がなければステップ407へ進み、矛盾があれば矛盾している実行必須条件を消去してステップ410へ進む。

【0021】ステップ407では、設定信号が指定する実行必須条件または実行優先度関数を、ICチップ12の記憶装置12mの条件テーブルに設定する。ステップ408では、前記ステップ407で実行優先度関数を設定した場合、他の機能の実行優先度関数と比較し関数値として上下関係が成立するものを発見し上下関係処理を行う。ここで、関数値として上下関係が成立するものとは、関数 f_1 、 f_2 が任意の入力に対して $f_1 \leq f_2$ となることをいう。また、上下関係処理とは、機能 A_1 、 A_2 の実行優先度関数 f_1 、 f_2 が関数値として上下関係が成立している場合には機能選択時には機能 A_1 が必ず選択されるようにした処理である。実行優先度関数が $f_1 \leq f_2$ を満たすということは、機能 A_1 が機能 A_2 よりも専門的であることを示している。そこで、どちらの機能も使用可能といった場合には、機能 A_1 の方が使用目的に近いと考えられるため、上下関係処理で機能 A_1 が選ばれるようにする。この上下関係処理については、図10を参照して後述する。ステップ409では、実行優先度関数値で選択できなかった場合に使用する実行優先順位を全て

の機能に対して付ける。実行優先順位は、例えば登録順に番号を付けて行く。そして、機能実行条件設定処理を終了する。

【0022】ステップ410では、実行不能信号を送信する。そして、機能実行条件設定処理を終了する。

【0023】図10は、上下関係処理(408)を示すフローチャートである。ステップ801では、前記ステップ407で設定した実行優先度関数(これを f_1 とする)を、他の機能の実行優先度関数(これを f_2 とする)と比較し、 $f_1 \leq f_2$ を満たす実行優先度関数 f_2 をもつ機能を探す。ステップ803では、 $f_1 \leq f_2$ を満たす実行優先度関数 f_2 が存在した場合はステップ804へ進み、存在しなかった場合はステップ805へ進む。ステップ804では、実行優先度関数 f_1 の機能に実行優先度関数 f_2 を加えると共にオリジナルな関数でないことを該当するオリジナルフラグに登録する。この操作を行うことにより、元の実行優先度関数が $f_1 \leq f_2$ ならば、新たな実行優先度関数(和)が $f_1 \geq f_2$ となり、前記ステップ308、309では、関数値和で選択するので、実行優先度関数 f_1 の機能が選ばれるようになる。ステップ805では、前記ステップ801～804をすべての他の機能に対して行う。

【0024】ステップ806では、前記ステップ407で設定した実行優先度関数(これを f_1 とする)を、他の機能の実行優先度関数(これを f_3 とする)と比較し、 $f_3 \leq f_1$ を満たす実行優先度関数 f_2 をもつ機能を探す。ステップ807では、 $f_3 \leq f_1$ を満たす実行優先度関数 f_3 が存在した場合はステップ808へ進み、存在しなかった場合はステップ809へ進む。ステップ808では、実行優先度関数 f_3 の機能に実行優先度関数 f_1 を加えると共にオリジナルな関数でないことを該当するオリジナルフラグに登録する。ステップ809では、前記ステップ806～808をすべての他の機能に対して行う。そして、上下関係処理を終了する。

【0025】図11は、アプリケーションプログラムの機能登録処理のフローチャートである。ステップ701では、ICチップ12の記憶装置12mに、新たなアプリケーションプログラムの機能を記憶する。ステップ702では、新たなアプリケーションプログラムの機能のすべての実行必須条件を記憶装置12mの条件テーブルに登録する。ステップ703では、新たなアプリケーションプログラムの機能の実行優先度関数の一つを記憶装置12mの条件テーブルに登録する。ステップ704では、図10の上下関係処理を行う。ステップ705では、新たなアプリケーションプログラムの機能の実行優先度関数で前記ステップ703、704を実行していないものが残っていれば前記ステップ703に戻り、残っていなければステップ706へ進む。ステップ706では、実行優先順位を全ての機能に対して付け直す。そして、アプリケーションプログラムの機能登録処理を終了

する。

【0026】図12は、アプリケーションプログラムの機能削除処理のフローチャートである。ステップ901では、ICチップ12の記憶装置12mから削除対象のアプリケーションプログラムの機能を削除する。ステップ902では、削除したアプリケーションプログラムの機能のすべての実行必須条件を記憶装置12mの条件テーブルから削除する。ステップ903では、削除したアプリケーションプログラムの機能の実行優先度関数の一つを記憶装置12mの条件テーブルから削除する実行優先度関数削除処理を実行する。この実行優先度関数削除処理については、図13を参照して後述する。ステップ904では、削除したアプリケーションプログラムの機能の実行優先度関数で削除していないものが残っていれば前記ステップ903に戻り、残っていなければステップ905へ進む。ステップ905では、実行優先順位を全ての機能に対して付け直す。そして、アプリケーションプログラムの機能削除処理を終了する。

【0027】図13は、実行優先度関数削除処理を示すフローチャートである。ステップ1001では、削除対象の実行優先度関数（これを f_1 とする）を記憶装置12mの条件テーブルから削除する。ステップ1002では、削除した実行優先度関数 f_1 のオリジナルフラグを参照し、オリジナルな関数であればステップ1003へ進み、オリジナルな関数でなければ実行優先度関数削除処理を終了する。ステップ1003では、記憶装置12mに記憶されている残りの機能の実行優先度関数の中に f_1 と同じ関数 f_1' が存在していないか調べる。ステップ1004では、関数 f_1' が存在し且つそれがオリジナルな関数であればステップ1005へ進み、関数 f_1' が存在しないか又は存在してもそれがオリジナルな関数でなければステップ1006へ進む。ステップ1005では、関数 f_1' を条件テーブルから削除する。ステップ1006では、 f_1 と同じ関数 f_1' が存在していないか調べ尽くしていなければ前記ステップ1003に戻り、調べ尽くしておれば実行優先度関数削除処理を終了する。

【0028】上記実施形態においては、ICカード1のICチップ12が実行必須条件の判定、実行優先度関数の判定、機能の選択を行うとしたが、リーダーライター側で行ってもよい。前者の方法は、ICカード1内の情報をリーダーライター側に渡す必要が無いため、情報の漏洩を防ぐことが出来る。一方、後者の方法は、判断を行うのがリーダーライター側であるため、ICカードの偽造に強い。ICカードの用途や重要度に応じて、これら2つの方法を適宜使い分けるのが良い。

【0029】また、上記実施形態において、例えば図4の実行優先度関数の内容の中に、

「10000円からは3%引き」

が存在するが、これを、

「 $f=0.03 * x$ if $x \geq 10000$, $f \geq 0$ otherwise, x : 利用金額」

というように、値引き額を算出する関数にしてもよい。

ただし、この場合には、関数によって値域が異なるの

で、機能の選択のときには新たな評価基準をとる必要がある。例えば、機能 A_1 , A_2 に対し値域が等しい関数 f_1 , f_2 を比較し、関数値が $f_1 \leq f_2$ であったならば A_2 に1加点し、最終的に最も多い点数の機能を選択するなどの処理が考えられる。

【0030】また、上記実施形態では、実行優先度関数値の和が最も高い機能が複数存在した場合、実行優先順位により機能を選択したが、利用者に問い合わせて選択させるようにしてもよい。実行優先度関数値の和が最も高い機能が複数存在する場合は稀であり、多くの場合は実行優先度関数値の和により機能を選択できるため、利用者に問い合わせて選択させるようにしても、従来より手続き処理時間を短縮できる。

【0031】以上のICカード1によれば、アプリケーションプログラムの各機能ごとに実行必須条件を設けているため、各機能ごとに実行を制限できる。すなわち、特定のアプリケーションプログラムや特定の機能の実行のみを制限できる。また、アプリケーションプログラムの各機能ごとに実行優先度関数や実行優先順位を設けているため、実行可能なアプリケーションプログラムやその機能が複数ある場合に、実行するアプリケーションプログラムやその機能の選択を自動化でき、手続き処理時間を短縮できる。また、実行必須条件および実行優先度関数ごとに変更フラグを設けているため、意図せぬ者による内容変更を禁止でき、セキュリティを確保できる。

【0032】さらに、実行必須条件をうまく組み合わせることにより、サービスの細分化が可能となる。すなわち、サービス提供者がアプリケーションプログラムを登録する際にICカード利用者の好みや要求を踏まえて実行必須条件や実行優先度関数を設定すれば、ICカード利用者の好みや要求に応じたサービスを提供できると共に余分なサービスを提供する無駄がなくなり、低料金のサービス提供が可能となる。

【0033】

【発明の効果】本発明のICカードのプログラム管理方法によれば、次の効果が得られる。

(1) ICカードの特定のアプリケーションプログラムまたはその一部の機能を制限する場合、従来はICカード全体が無効化されてしまったが、本発明では当該アプリケーションプログラムまたは機能だけを制限できるようになり、他のアプリケーションプログラムまたは機能は無駄にせずに利用できる。

(2) 実行可能なアプリケーションプログラムまたは機能が複数存在した場合、従来は利用者による選択が必要であったが、本発明では自動選択できるようになり、手続き処理時間を短縮できる。

(3) 実行条件または実行優先度を定める情報を変更しようとした場合、変更フラグにより変更可否をチェックするため、意図せぬ者による内容変更を禁止でき、セキュリティを確保できる。

【図面の簡単な説明】

【図1】本発明の一実施形態のICカードの斜視図である。

【図2】図1のICカード内部のICチップの構成を示すブロック図である。

【図3】図2のICチップに記憶された条件テーブルの構成説明図である。

【図4】図2のICチップに記憶された条件テーブルの続きの構成説明図である。

【図5】図2のICチップに記憶された条件テーブルの更に続きの構成説明図である。

【図6】図1のICカードのメイン処理を示すフローチャートである。

【図7】図1のICカードの実行必須条件チェック処理

を示すフローチャートである。

【図8】図1のICカードの機能選択実行処理を示すフローチャートである。

【図9】図1のICカードの機能実行条件設定処理を示すフローチャートである。

【図10】図1のICカードの上下関係処理を示すフローチャートである。

【図11】図1のICカードのアプリケーションプログラムの機能登録処理を示すフローチャートである。

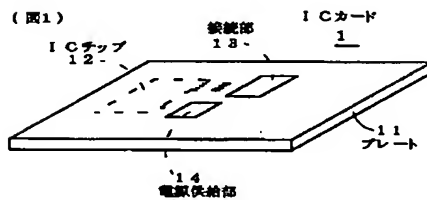
【図12】図1のICカードのアプリケーションプログラムの機能削除処理を示すフローチャートである。

【図13】図1のICカードの実行優先度関数削除処理を示すフローチャートである。

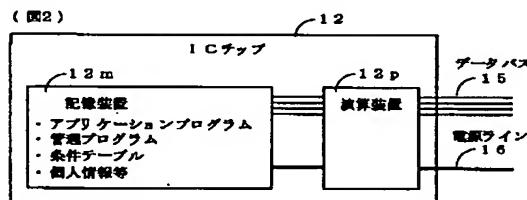
【符号の説明】

1…ICカード、11…プレート、12…ICチップ、12m…記憶装置、12p…演算装置、13…接続部、14…電源供給部。

【図1】



【図2】



【図3】

アプリケーション名	機能名	実行条件設定			
		利用者に よる変更	サービス提供者 による変更	オリジナル	内容
〇〇クレジットカード	支払い	×	○	○	〇月×日まで有効
	入金	×	○	○	1999年10月まで入金可能
△△電子マネー	引き出し	○	○	○	一度に1000円まで
	引き入れ	○	○	○	無条件
□□クレジットカード	支払い	×	○	○	一度に10000円まで
××電算機	電算機	×	○	○	〇月〇日まで有効

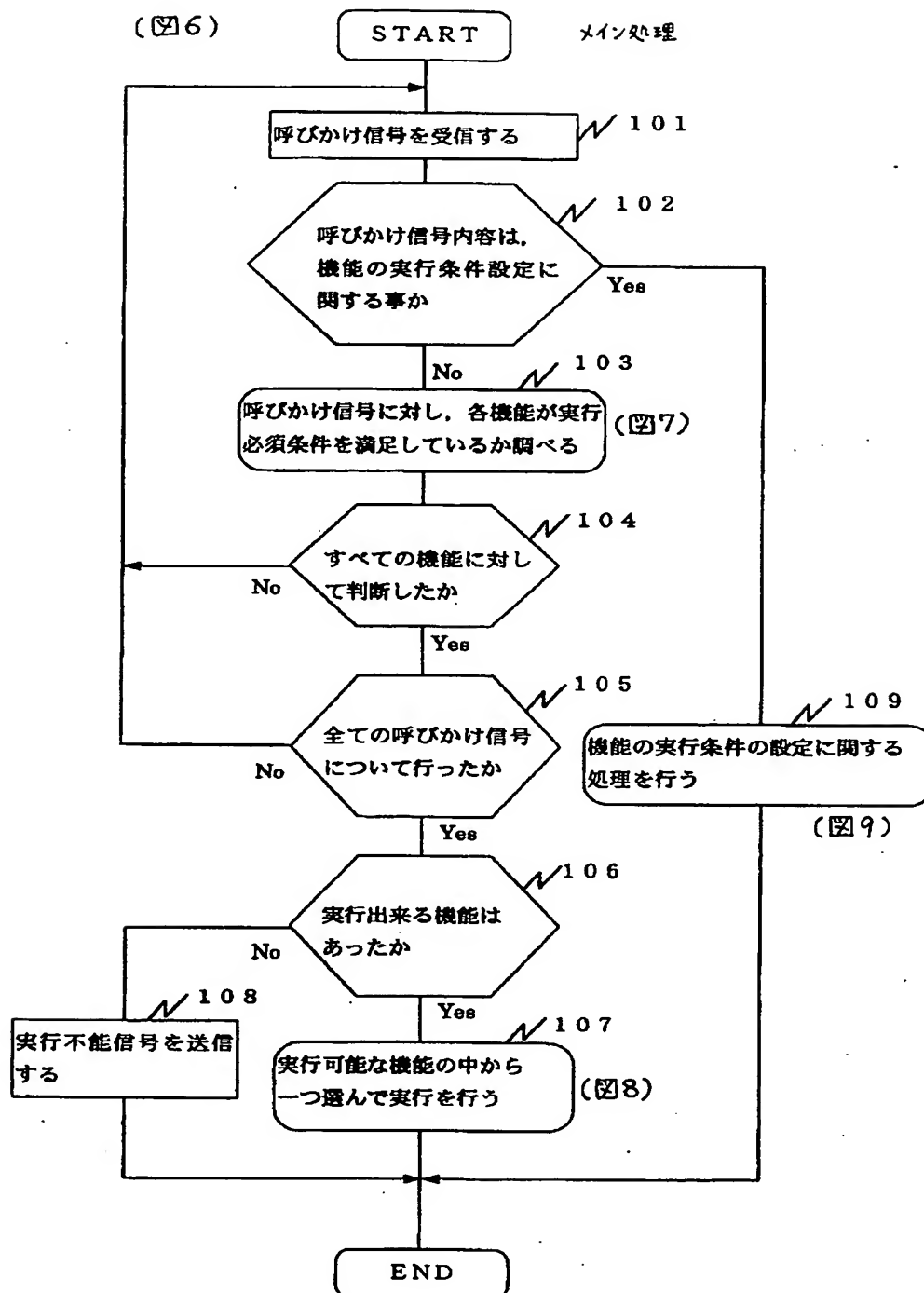
【図4】

アプリケーション名	機能名	実行条件設定			
		利用者に よる変更	サービス提供者 による変更	オリジナル	内容
〇〇クレジットカード	支払い	○	×	Yes	〇〇月以内での利用
	入金	○	○	Yes	
△△電子マネー	引き出し	○	×	Yes	一度に1000円まで
	引き入れ	○	○	Yes	
□□クレジットカード	支払い	×	○	Yes	1999年10月より引き金
××電算機	電算機	○	×	Yes	A国—C国間での利用

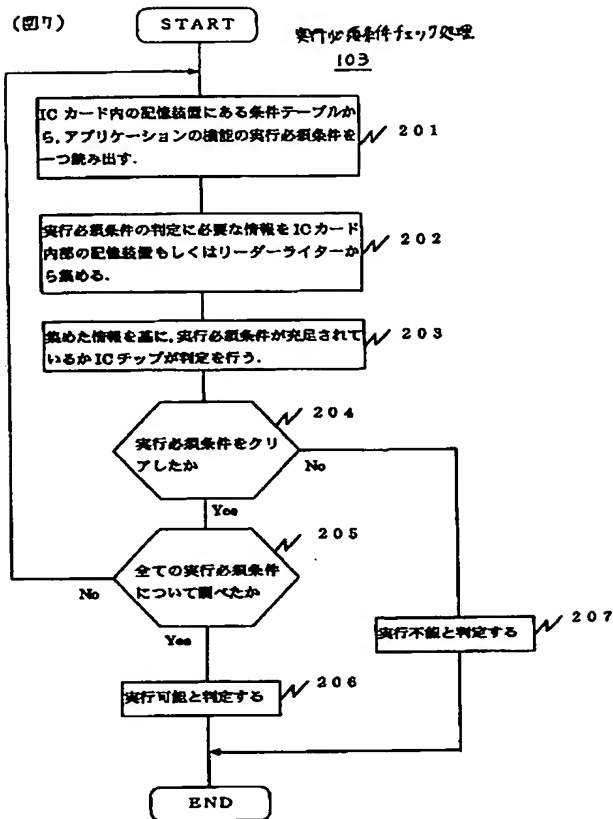
【図5】

アプリケーション名	機能名	実行条件設定A				実行条件 設定
		利用者に よる変更	サービス提供者 による変更	オリジナル	内容	
〇〇クレジットカード	支払い	○	×	Yes	〇〇月までは ×国利用での 利用	1
△△電子マネー	入金	○	○	Yes		2
	引き出し	○	×	Yes		3
	引き入れ	○	○	Yes		4
□□クレジットカード	支払い	×	○	Yes		5
××電算機	電算機	○	×	Yes		6

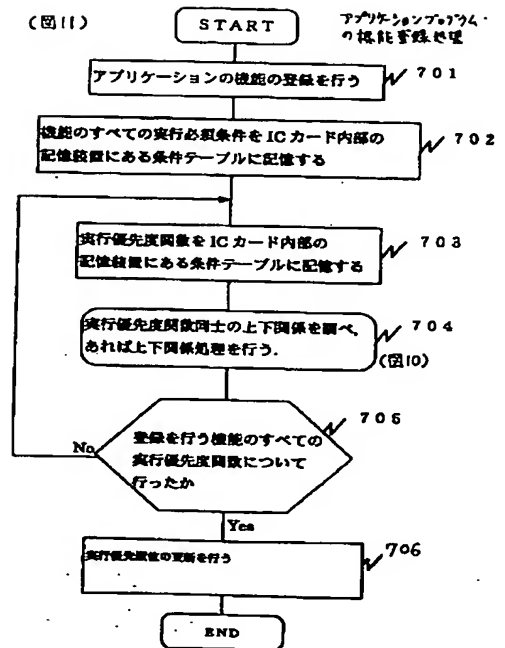
【図6】



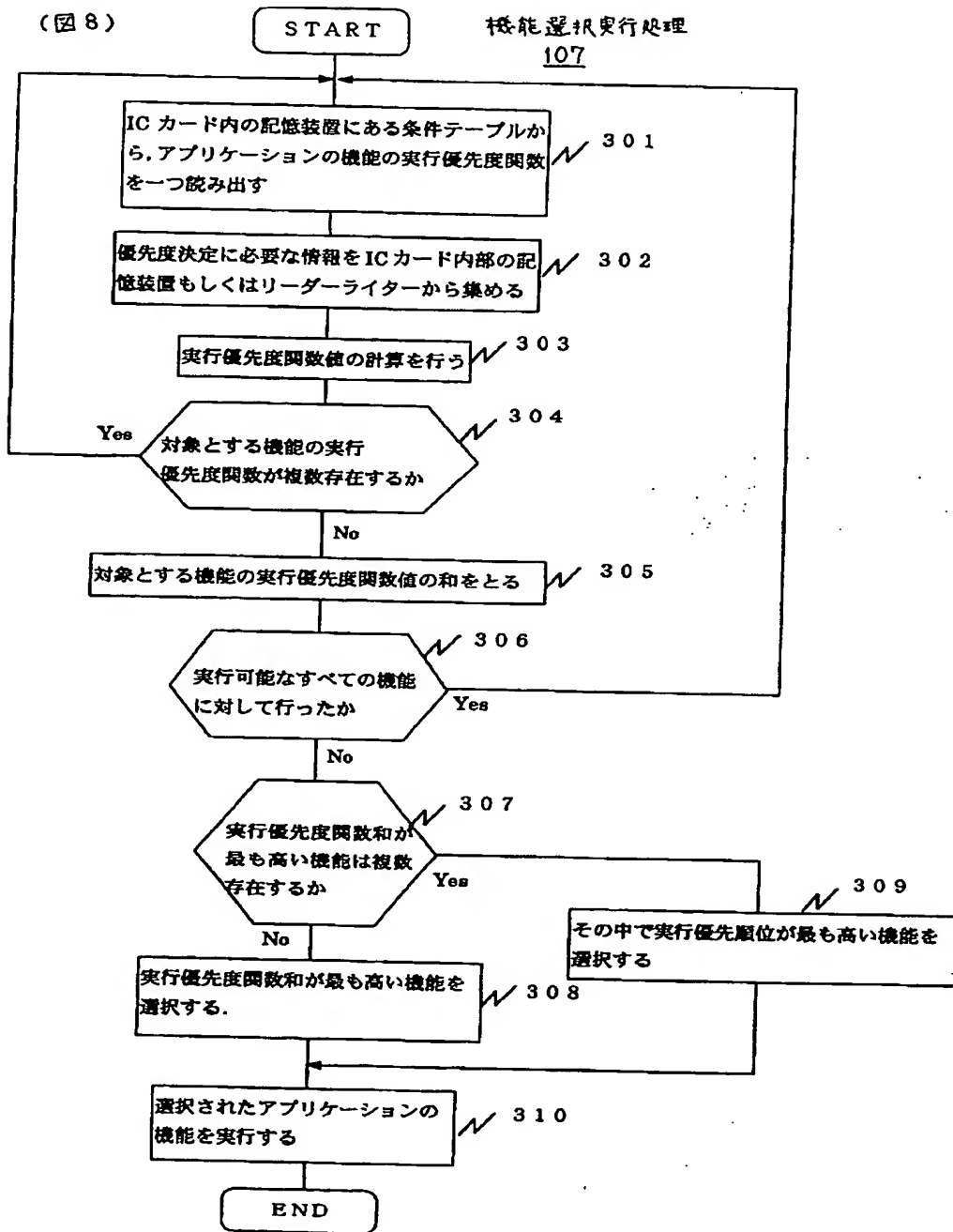
【図7】



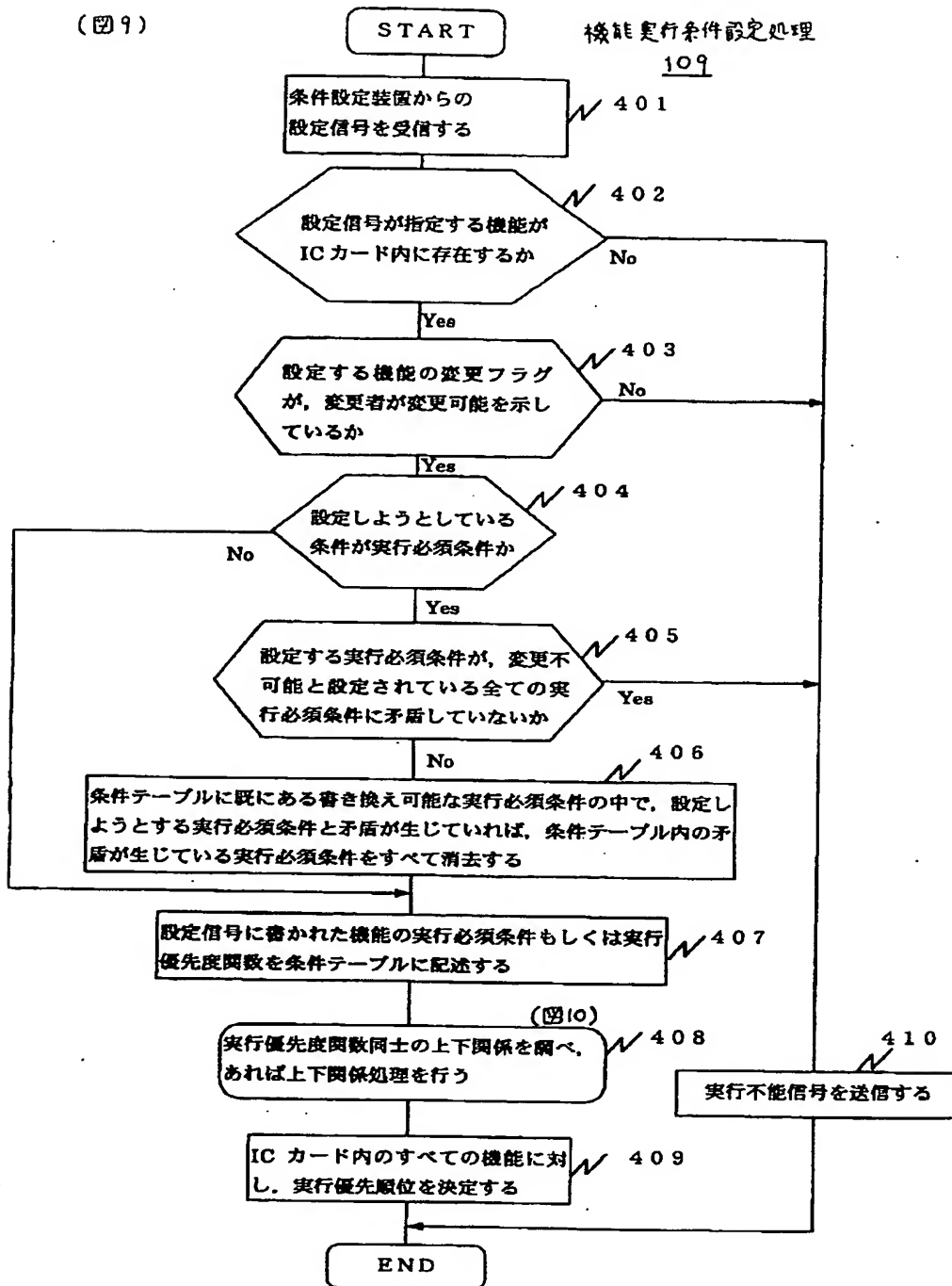
【図11】



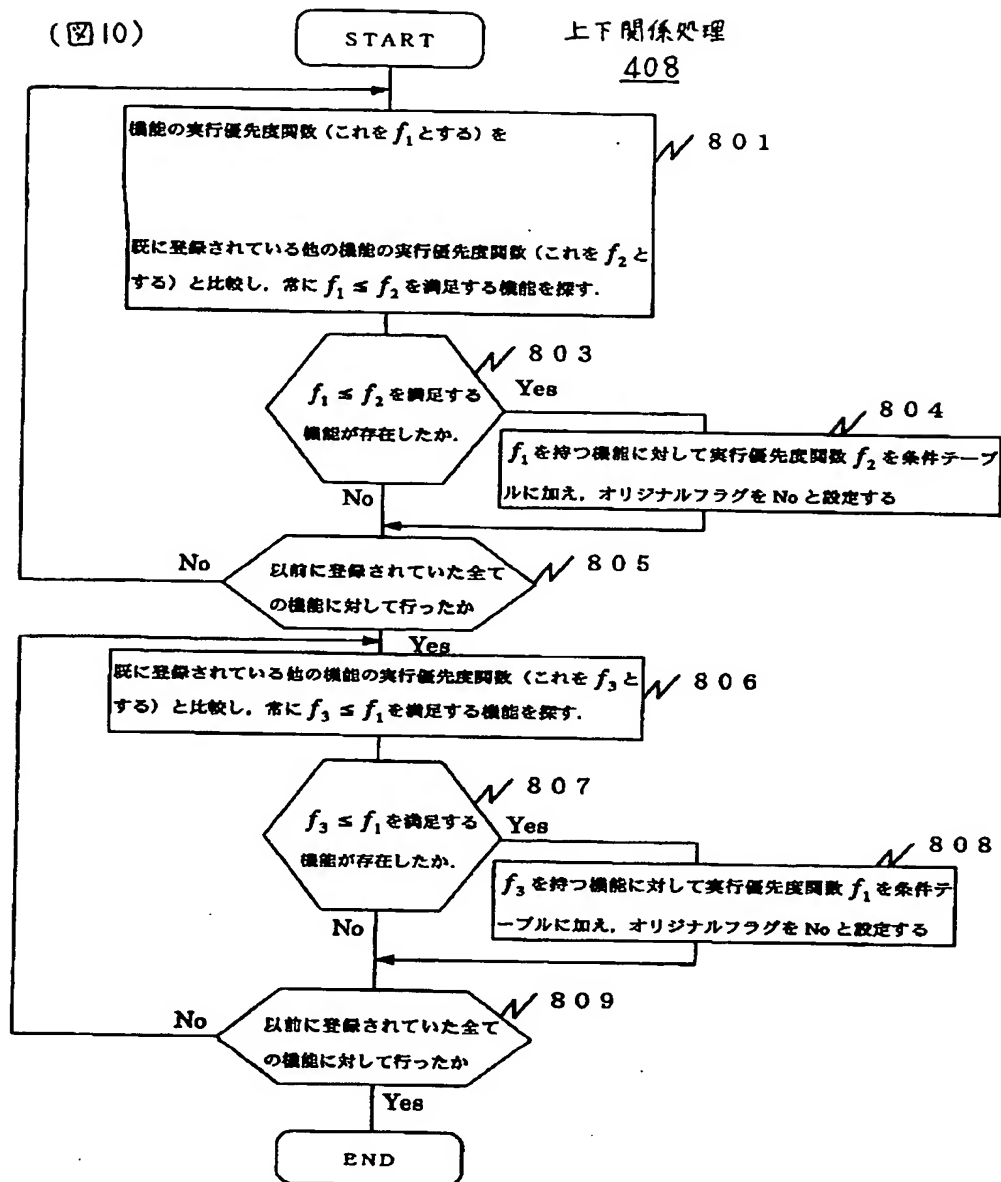
【図8】



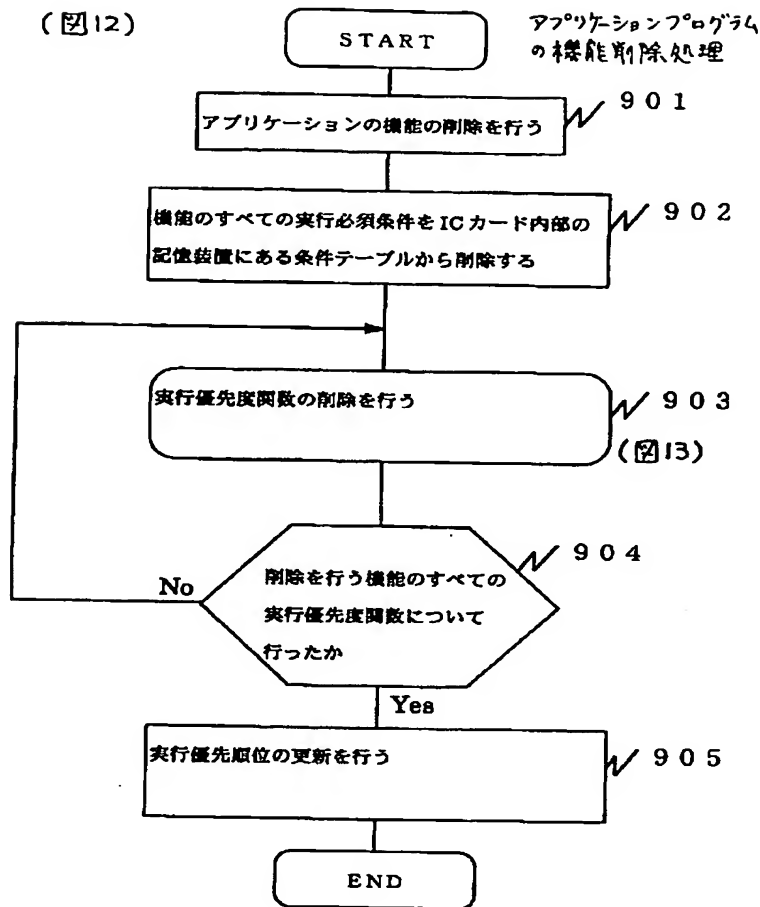
【図9】



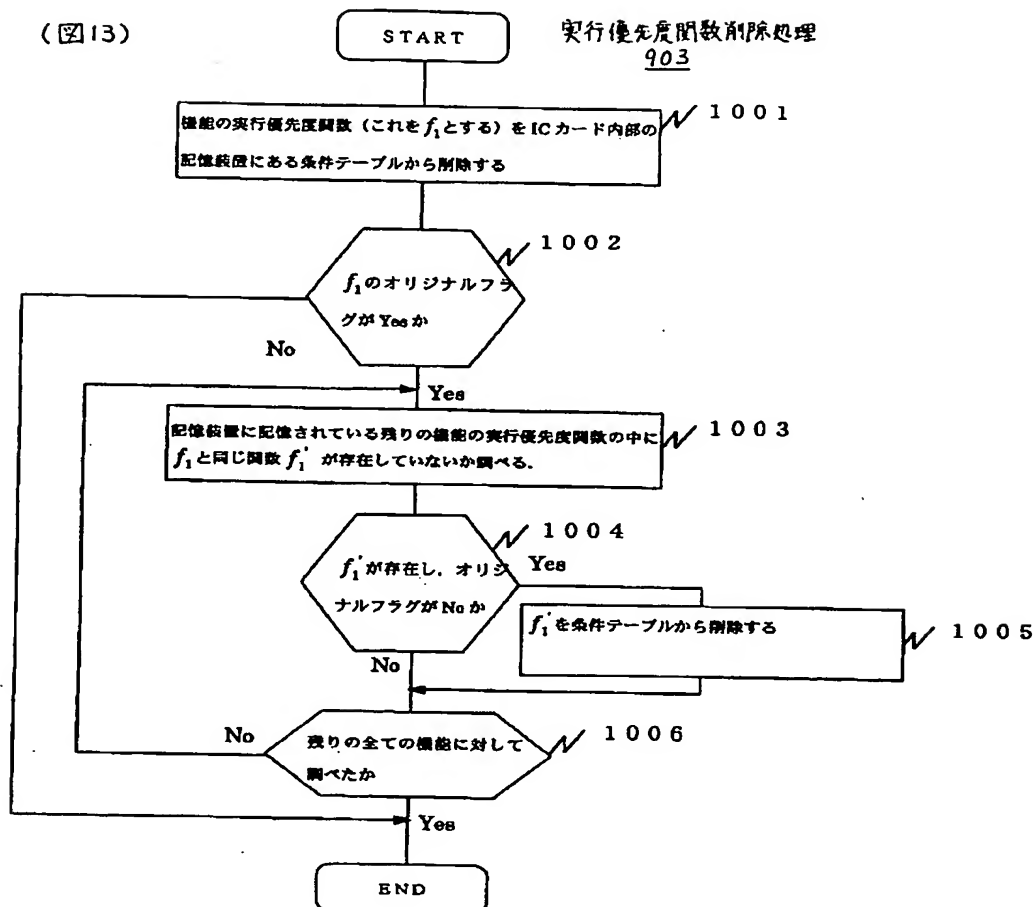
【図10】



【図12】



【図13】



フロントページの続き

(72)発明者 佐々木 敏郎
神奈川県川崎市麻生区王禅寺1099番地 株
式会社日立製作所システム開発研究所内

Fターム(参考) 5B035 AA06 BB09 BB11 BC00 CA11
5B058 CA25 CA28 YA03 YA20
5B076 AB17 CA07